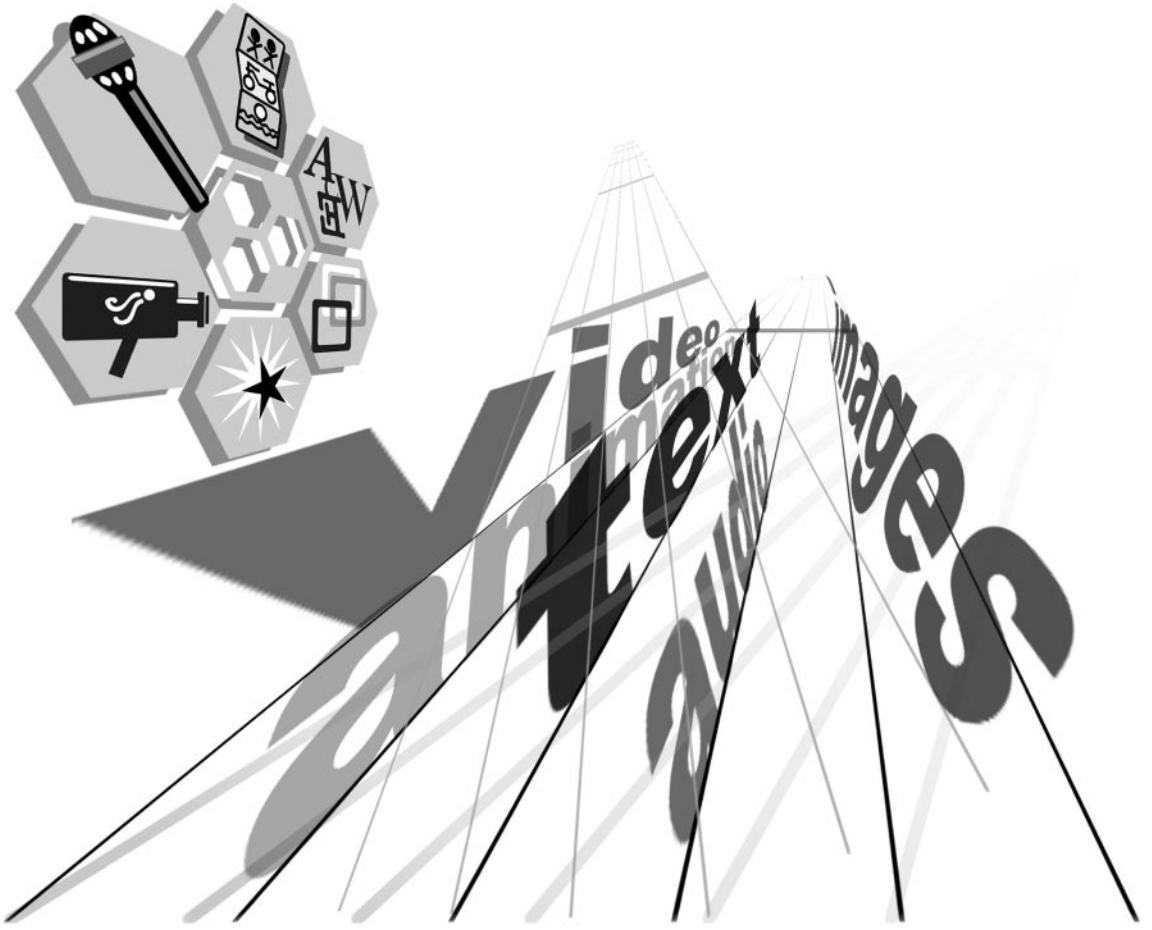
AUTHORING CONTENT FOR REALPLAYER® 7

RealNetworks Technical Blueprint Series

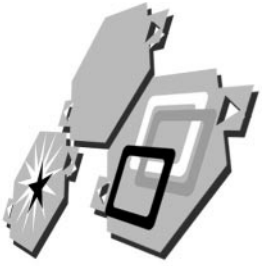
29 October 1999



RealNetworks, Inc.
2601 Elliott Avenue
Seattle, WA 98121 USA

http://www.real.com

CONTENTS



AUTHORING CONTENT FOR REALPLAYER 7

New features in RealPlayer 7 give you greater flexibility in designing streaming media Web sites:

- You can instruct RealPlayer 7 to cache image files downloaded through HTTP. In contrast, RealPlayer G2 does not cache any files on disk.
- You can write hyperlinks that open content in new RealPlayer 7 windows. RealPlayer G2, however, supports only one display window.

Although these features are not backwards compatible with RealPlayer G2, you can use the SMIL `<switch>` statement to provide alternate presentations for RealPlayer 7 and RealPlayer G2. The RealPlayer 7 features work for content served from a Web server, RealServer G2 Gold, or RealServer G2 6.1.

Additional Information

To use these features, you need to be familiar with SMIL, which is described in *RealSystem G2 Production Guide*, and RealText, which is documented in *RealText Authoring Guide*. Both documents are available for download from **http://service.real.com/help/library/index.html**.

HTTP Caching

RealPlayer 7 can cache on disk any files delivered through HTTP. You may want to cache GIF or JPEG images that are part of a SMIL presentation, for example. Caching works only for files delivered through HTTP, and you should not cache clips more effectively served through RTSP. These include video, audio, RealPix, Flash, and RealText clips. Nor should you cache images such as ads or any files that do not consistently appear each time the presentation is played.

Note

Disk caching applies only to images listed in SMIL files, not to JPEG and GIF files used in a RealPix clip.

RealPlayer caches RealPix images in memory, but not on disk, for the duration of the RealPix presentation. You should not try to cache on disk the images in a RealPix presentation.

Caching images is beneficial if your viewers return to your streaming presentation frequently. An example is an Internet radio station that uses GIF logos and on-screen buttons to create a branded layout whenever the station streams to RealPlayer. As long as the GIFs reside in the RealPlayer cache, the server does not have to resend the files on each of RealPlayer's subsequent visits. As a result, the presentation launches faster when viewers revisit.

RealPlayer caches files within the RealPlayer home directory in a folder named `cache_db`. Its cache is independent of any Web browser cache. The default RealPlayer cache size is 4 MB, though the user can change the size. The cache is small because it is meant to hold small images such as buttons and logos, not large graphics or streaming clips.

The "CHTTP" Caching Protocol

RealPlayer does not cache all items streamed by HTTP. Instead, you designate files to cache by using `chttp://` instead of `http://` in the file's URL. When RealPlayer reads a CHTTP URL in a SMIL file, it first checks its disk cache for the file. If the file isn't present, it requests the file through HTTP, storing the file in its cache. Because RealPlayer interprets a `chttp://` URL as a special instance of normal HTTP downloading, caching works for any file stored on an HTTP-compatible server.

If that same RealPlayer subsequently requests another presentation that includes the cached file in a CHTTP URL, the cached version is used. The cached version is not used, though, if the file URL is regular HTTP or differs in any way from the original CHTTP URL. As well, any new file requested through RTSP, HTTP, or PNA is not cached. The following SMIL example indicates that the GIF image should be initially downloaded, then cached for later use:

```
<img src="chttp://www.company.com/images/image1.gif" region="image"/>
```

This URL causes RealPlayer to store `image1.gif` in the cache for later use until it has expired or is overwritten. "Cache Size and Expiration Rules" on page 3 describes the cache expiration rules.

Controlling the RealPlayer 7 Cache

RealPlayer supports the same HTTP header fields used to control file expiration in Web browser caches. It can thereby carry out caching directives set by Web servers. This lets you reuse Web page images in RealPlayer presentations without losing control of how these images are cached. The following sections describe how to use HTTP headers to control the RealPlayer cache, and how RealPlayer manages its cache.

Additional Information

Documentation for most Web servers includes information about how to set fields in HTTP header files.

Overriding Caching with Cache-Control

The `Cache-Control` command of an HTTP header file can override caching of a RealPlayer file requested through `chttp://`. A file requested through CHTTP is not cached if any of the following are present as meta-information in the HTTP header file:

- `Cache-Control:no-cache`
- `Cache-Control:no-store`
- `Cache-Control:private`
- `Cache-Control:must-revalidate`

Cache Size and Expiration Rules

By default, RealPlayer's HTTP cache size is set to 4 MB, although users can modify this size. Unless an HTTP header sets a file lifetime, the cached file expires after 4 hours. A subsequent request for a cached item restarts the item's expiration clock. When the cache fills, RealPlayer begins to delete unexpired items to reclaim needed disk space on a first in, first out basis.

Changing the Lifetime of a Cached File

Within an HTTP header, `Cache-Control:max-age` can set the "time to live" (TTL) for a cached file, overriding the default expiration time. Specified as seconds, the maximum age is added to the current time to get the file's expiration time. This value must be between 60 seconds and one year. For example:

```
Cache-Control:max-age=172800
```

If the `Cache-Control:max-age` field is not used, an `Expires` field can determine expiration time. The `Expires` field takes as an attribute a date string that defines when the cached element expires, relative to the caching computer's clock. The date string is formatted as:

`Expires= Wdy, DD Mon YYYY HH:MM:SS GMT`

The weekday is optional. Here are two examples, the first with the weekday designation and the second without:

```
Expires= Fri, 17 Mar 2000 19:37:09 GMT
```

```
Expires= 17 Mar 2000 19:37:09 GMT
```

The weekday and month abbreviations are the following:

day of week:	Mon, Tue, Wed, Thu, Fri, Sat, Sun
month:	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

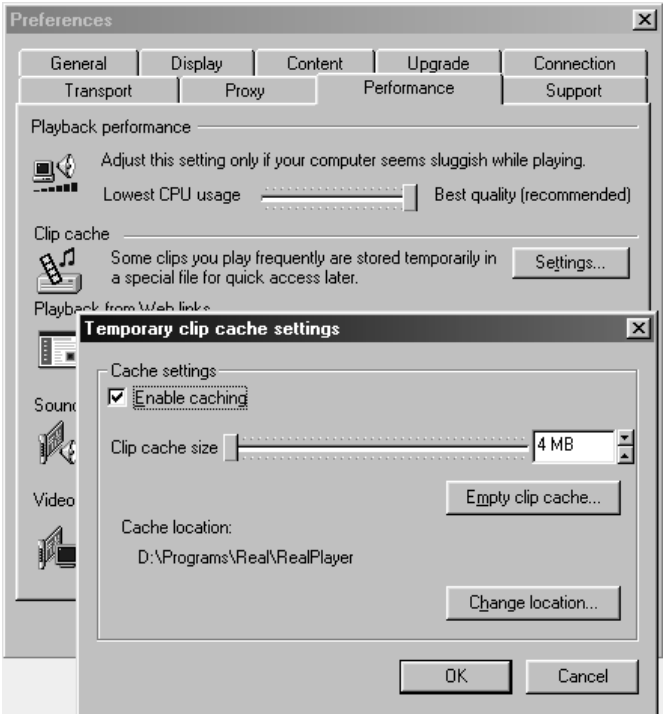
Note

The entry is not cached if the value of the `Expires:` field is less than the current date and time.

User Control of the RealPlayer 7 Cache

RealPlayer users can control some aspects of RealPlayer's cache by choosing **View>Preferences** and clicking the **Performance** tab. The user then clicks **Settings...** under **Clip cache**. The following figure illustrates the cache control dialog.

RealPlayer Cache Control Dialog



The RealPlayer 7 cache control dialog uses the following fields and buttons:

Enable caching

Caching is set on by default. The user can disable it by unchecking the **Enable caching** box. Disabling the cache does not clear it. Cached items are cleared according to the expiration rules described above, or until the cache is emptied as described below.

Cache size

Moving the slider or typing a new value in the text box changes the disk cache size. The minimum value is 1 MB. The maximum is 1 GB or the maximum size of the target drive, whichever is smaller.

Empty clip cache...

By clicking this button and confirming the action, the user can empty the cache of all content.

Change disk location

This feature is not yet functional.

Authoring SMIL Files for Caching

When you cache files, author your SMIL file to download the cached items before other streamed elements. You can do this by placing the cached elements in a SMIL <seq> group head of the streamed elements. The following SMIL extract, which omits the header information for convenience, caches two logo images. The SMIL file is for a presentation in which logos stay the same, but the streaming RealVideo and RealText clips change frequently:

```
<smil>
  ...header omitted from example...
  <body>
    <seq>
      <!-- Download and cache these two logos. -->
      <img src="chttp://www.company.com/images/logo1.gif?bitrate=20000"
        region="logo1" fill="freeze"/>
      <img src="chttp://www.company.com/images/logo2.gif?bitrate=20000"
        region="logo2" fill="freeze"/>
      <par>
        <!--Next, stream these 2 clips in parallel. -->
        <textstream src="rtsp://realserver.company.com/news.rt"
          region="newsregion" fill="freeze"/>
        <video src="rtsp://realserver.company.com/newsvid.rm"
          region="videoregion"/>
      </par>
    </seq>
  </body>
</smil>
```

In this example, the two logos quickly download in sequence. Next, RealServer streams a parallel group comprising the RealVideo and RealText clips. When RealPlayer plays this presentation again, it first checks its cache for the two logos. If it finds them, it skips directly to the streaming clips.

Additional Information

For information about authoring SMIL, see the SMIL chapter in *RealSystem G2 Production Guide*, available at **http://service.real.com/help/library/index.html**.

Multiple RealPlayer 7 Windows

RealPlayer 7 can open as many player windows as the host CPU and memory allows. This lets you keep navigation information visible in one window, for

example, while content plays in another window. New windows open when a viewer clicks a specially configured hyperlink within a SMIL or RealText file. They cannot be launched automatically.

Additional Information

For basic information about authoring, see the SMIL chapter in *RealSystem G2 Production Guide*. *RealText Authoring Guide* explains the RealText mark-up. Both manuals are available for free download at **http://service.real.com/help/library/index.html**.

Creating a Link for a New Window

You open a new RealPlayer window through a hypertext link in a RealText or SMIL file. RealText and SMIL both support HTML-style hypertext links. Unlike HTML, RealSystem mark-up tags are case-sensitive in accordance with their XML-based syntax. A RealSystem hyperlink that opens a new RealPlayer window uses this format:

```
<a href="command:openwindow(name, URL)">...</a>
```

When a viewer clicks a link with this syntax, `command:openwindow` tells RealPlayer to open a new window for a given URL, and stop the presentation on the current window. This command takes two arguments, *name* and *URL*. Either argument may be quoted. You can separate arguments with a comma, but it is not required. A space may precede or follow the comma.

name

The *name* argument gives a user-defined name for the new Player window. The following are the predefined values and rules for the *name* argument:

- If *name* is `_new` or `_blank`, a new player window opens each time the viewer clicks the link.
- If *name* is `_current` or `_self`, the URL opens in the same player window as the link.
- Any other `name` value creates a player window with that name. A subsequent `openwindow` command using the same name opens the given URL in the same window.

URL

The *URL* argument is the fully qualified URL to the clip or SMIL presentation opened in the new window. Relative URLs do not work. You must include the protocol, such as rtsp://, http://, chttp://, or file:// in the URL.

Examples of Opening New RealPlayer Windows

The following sections provide examples of how to create hyperlinks that launch new RealPlayer windows. The first three sections illustrate how to target various windows with the hyperlink syntax. They use RealText links as examples. The fourth example shows a link created from an image in a SMIL presentation. The fifth example shows a SMIL hotspot link for a video clip.

Targeting the Same Window with Multiple Links

The following RealText link opens a URL in a new RealPlayer window named feature:

```
<a href="command:openwindow(feature,
rtsp://realserver.company.com/comedy.rm)">Comedy Hour</a>
```

When first clicked, this link creates a RealPlayer window named feature. If another link also targets the feature window, clicking that link starts the new URL in the feature window. Clicking the following link, for example, starts an animal program in the window running the comedy program:

```
<a href="command:openwindow(feature,
rtsp://realserver.company.com/animals.rm)">Sharks!</a>
```

Opening Separate Windows

Each link opens a separate window if the window names are different, or you use the predefined name _new or _blank:

```
<a href="command:openwindow(_new,
rtsp://realserver.company.com/comedy.rm)">Comedy Hour</a>
```

```
<a href="command:openwindow(_blank,
rtsp://realserver.company.com/animals.rm)">Sharks!</a>
```

Launching URLs in the Current Window

Use either _current or _self to open the URL in the current window:

```
<a href="command:openwindow(_self,
rtsp://realserver.company.com/comedy.rm)">Comedy Hour</a>
```

```
<a href="command:openwindow(_current,
rtsp://realserver.company.com/animals.rm)">Sharks!</a>
```

Linking from a SMIL File Image

A SMIL file cannot display text on screen, so it does not support the text links described above for RealText. A SMIL file can make hyperlinks out of GIF and JPEG images, however. You can use an image file as a button, for example, that launches a new RealPlayer window. The following sample shows a hyperlinked image file within a SMIL file. The hyperlink syntax is the same as described above for RealText:

```
<a href="command:openwindow(_new, rtsp://realserver.company.com/animals.rm)">
  <img src="launch_button.gif" region="button"/>
</a>
```

Tip

You can make a hyperlink out of any clip within a SMIL file, not just image files.

Creating a SMIL Hotspot Link

SMIL lets you create hotspot links (image maps) over any clip. The SMIL chapter in the *RealSystem G2 Production Guide* explains the hotspot link syntax. The following example defines a hotspot link over the upper, left-hand quarter of a RealVideo file. Clicking this hotspot launches the URL in a new RealPlayer window:

```
<video src="rtsp://realserver.company.com/video/video1.rm" region="video">
  <anchor rtsp="command:openwindow(_new,
    rtsp://realserver.company.com/video/video2.rm)" coords="0,0,25%,25%"/>
</video>
```

RealPlayer Version Checking

RealPlayer G2 cannot play presentations that use the caching and multiple window features of RealPlayer 7. However, you can create two presentations, one for RealPlayer 7 and one for RealPlayer G2. You use the same streaming clips, but author different RealText files and use the SMIL `<switch>` tag to let each RealPlayer pick the right presentation. To facilitate this choice, RealPlayer 7 supports a new SMIL `<switch>` tag test attribute: `system-required`.

Using <switch> to Support Different Versions of RealPlayer

To let RealPlayer 7 and RealPlayer G2 choose the clips they can play, you write a SMIL file that has one or more <switch> tags. Each <switch> tag offers two choices. The first choice has a system-required attribute that indicates RealPlayer 7 is required. The second choice, which is meant for RealPlayer G2, does not have a system-required attribute. Each RealPlayer then evaluates the <switch> tag and determines which set of clips it can play. A SMIL file with this type of <switch> tag uses this general form:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
 <body>
  <switch>
   <seq system-required="cv" cv:systemComponent="...attribute to test...">
    ...clips to play if RealPlayer satisfies the test attribute...
   </seq>
   <seq>
    ...default choice...
   </seq>
  </switch>
 </body>
</smil>
```

The sample <switch> tag above lets RealPlayer choose between two <seq> groups, but RealPlayer can choose between <par> groups and individual clips as well. The final, default choice must always be included and must not use the system-required test attribute:

- Unlike RealPlayer 7, RealPlayer G2 ignores <switch> tag choices with the system-required test attribute. Hence there must be one <switch> choice that does not include the attribute.
- Each RealPlayer evaluates <switch> tag choices in order, choosing the first one it can play. If the default choice with no system-required test attribute came first, RealPlayer 7 would choose it before evaluating the next choice. Hence the default choice must always be last in the <switch> tag.

Additional Information

The chapter on SMIL in the *RealSystem G2 Production Guide* explains the basics of authoring a <switch> statement.

Note

To test a <switch> tag, you'll need to have two computers available because you cannot run both RealPlayer 7 and RealPlayer G2 on the same machine.

Writing a <switch> Tag

There are two aspects to writing a <switch> tag with a system-required test attribute:

1. Declaring the proper XML namespace in the <smil> tag.
2. Writing the <switch> tag with the proper system-required and systemComponent syntax.

Declaring the XML Namespace

Although RealPlayer's implementation of SMIL 1.0 does not typically need a XML namespace declaration, the use of systemComponent as a test attribute requires the following declaration:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
```

The namespace must be declared exactly as shown above. This declaration informs RealPlayer 7 that the SMIL file uses the XML extension systemComponent. The declaration has the following components:

xmlns:	Indicates an XML namespace declaration.
cv=	Namespace name defined by RealNetworks. The "cv" stands for "component version."
"*http://...*"	The quoted URL "http://features.real.com/systemComponent" is used by RealPlayer solely as an identifier that uniquely defines the namespace. RealPlayer does not contact this URL. The URL is valid, however, and displays an HTML page that describes the feature.

Defining a <switch> Tag Test Attribute

Within the <switch> tag, you include a system-required test attribute has this form:

```
system-required="cv"
cv:systemComponent="http://features.real.com/?feature;player=7.0.x"
```

The components of the test attribute are as follows:

`system-required="cv"`	Refers to the "component version" namespace defined in the `<smil>` tag.
`cv:systemComponent=`	Defines a new feature test attribute. Case must be exact.
"*http://...*"	The quoted URL "`http://features.real.com?feature;`" is the unique identifier for the component syntax. It matches the URL in the `<smil>` tag namespace declaration. RealPlayer does not request this URL.
`player=`	Specifies that `systemComponent` checks for a RealPlayer feature.
`7.0.x`	Indicates version of RealPlayer to check for, such as 7.0.23. All RealPlayers with the given version number or higher choose this test attribute. You can find a RealPlayer's version number with the **About RealPlayer** command under the **Help** or **Apple** menu.

Note

The RealPlayer version, such as 7.0.23, is the only part of the `systemComponent` syntax you should change.

Example 1: Caching

The following example is a SMIL `<switch>` tag that causes RealPlayer 7 to download and cache two logo files. If RealPlayer G2 requests this SMIL file, it simply downloads the logo files. This extract revises the SMIL file described in "Authoring SMIL Files for Caching" on page 6. Bolding is used only to emphasize crucial aspects of the example:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
 <body>
  ....
  <switch>
   <seq system-required="cv"
    cv:systemComponent="http://features.real.com/?feature;player=7.0.23">
    <!-- For RealPlayer 7, download and cache these two logos" -->
    <img src="chttp://www.company.com/images/logo1.gif?bitrate=20000"
     region="logo1" fill="freeze"/>
    <img src="chttp://www.company.com/images/logo2.gif?bitrate=20000"
     region="logo2" fill="freeze"/>
   </seq>
   <seq>
    <!-- For RealPlayer G2, download these two logos" -->
```

```
      <img src="http://www.company.com/images/logo1.gif?bitrate=20000"
        region="logo1" fill="freeze"/>
      <img src="http://www.company.com/images/logo2.gif?bitrate=20000"
        region="logo2" fill="freeze"/>
    </seq>
   </switch>
  </body>
</smil>
```

When evaluating this <switch> statement, RealPlayer 7 chooses the first <seq> group. For the actual version number, shown above as7.0.23, use the version number from your RealPlayer. RealPlayer G2 ignores the first <seq> group and chooses the second <seq> group, which has no test attribute.

Example 2: Multiple Windows

The following SMIL <switch> tag presents RealPlayer with a choice between two RealText files. The first RealText file has hyperlinks that launch clips in new RealPlayer windows. The second RealText file is written for RealPlayer G2. It plays clips in the main RealPlayer window:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
 <body>
  ....
  <switch>
   <!-- RealText clip to play with RealPlayer 7. -->
   <textstream system-required="cv"
     cv:systemComponent="http://features.real.com/?feature;player=7.0.23"
      src="rtsp://realserver.company.com/realtext/navigateRP7.rt
      region="text" fill="freeze"/>
   <!-- RealText clip to play with RealPlayer G2. -->
   <textstream
     src="rtsp://realserver.company.com/realtext/navigateG2.rt
     region="text" fill="freeze"/>
  </switch>
 </body>
</smil>
```